Robust Non-Interactive Zero-Knowledge Watermarking Scheme Against Cheating Prover

Chia-Mu Yu and Chun-Shien Lu*
Institute of Information Science, Academia Sinica
Taipei City, Taiwan 115, Republic of China

ABSTRACT

Most watermarking methods presented so far belong to the category of symmetric watermarking in that the secret key is undesirably revealed during the watermark detection process. In view of this security leakage, zero-knowledge watermark detection (ZKWD) has been introduced without obviously revealing the secret information. However, the existing ZKWD protocols still suffer from some challenging problems, which will be addressed in this paper. First, a zero-knowledge watermark detection protocol is presented based on our robust image watermark scheme so that robustness against removal and geometric attacks can still be retained. The aim is to extend the capability of our own system to satisfy more watermarking requirements as possible. Second, the watermarks revealed for zero-knowledge detection are still secured by a secret key-based shuffling function so that the verifier cannot have the knowledge about the hidden watermarks. Third, we show how our protocol can be operated in a non-interactive way. Finally, in order to prevent from cheating prover, the Hamiltonian cycle is introduced in a media data before its publication. Thus, the overall characteristics distinguish our protocol from the existing protocols significantly.

Categories and Subject Descriptors

K [.]: 4.4 [Electronic Commerce]: *Security and Intellectual property*; K.6.5 [Management of Computing and Information Systems]: *Security and Protection*

General Terms

Algorithms, Security

Keywords

Ambiguity attack, Robustness, Security, Watermark, Zero-knowledge detection

*Corresponding author: Dr. C. S. Lu (email: lcs@iis.sinica.edu.tw)

MM-SEC'05, August 1–2, 2005, New York, New York, USA

1. INTRODUCTION

1.1 Background

Digital watermarking has been recognized as a helpful technology for copyright protection, traitor tracing, and authentication, where robustness is considered to be a critical issue affecting the practicability of the watermarking system and has received most attention. The basis of conventional watermarking methods is established based on Kerckhoffs principle [13] in that the watermarking embedding and detection algorithms are public and the parameters (e.g., secret key) are kept secret. This implies that the secrecy of a watermarking system relies on the secret key instead of the algorithm. As a result, watermarking methods presented to date mostly obey this principle; i.e., the embedding and the detection keys are the same and kept secret. This watermarking paradigm is called "symmetric" watermarking.

Basically, when the secret key in symmetric watermarking is to be used for detection, it is exposed accordingly and can merely be used only once. In addition, the disclosure of the secret key can efficiently assist watermark-estimation attack [14] in removing watermarks. One solution to this problem is to use a secret key for embedding and a different but public key for watermark detection. This is known as "asymmetric" watermarking in which Hartung and Girod's study [12] is believed to pioneer this topic. However, the current asymmetric watermarking approaches only possess limited robustness [9] or suffer from the difficulty of securely integrating the asymmetric protocol and other watermarking components [11].

An alternative way in investigating a symmetric watermark detection protocol without revealing the secret information is zero-knowledge watermark detection [1, 2, 3, 7, 8]. The principle behind the zero-knowledge detection protocol is that a prover able to convince a verifier that he/she certainly knows a secret without revealing this secret to verifier. The first zero-knowledge watermarking scheme was introduced by Craver [7]. In this protocol, the prover generates many fake watermarks and combine these with the legal watermark into a set, and then prove to verifier that there is a legal watermark belongs to this set. The above procedures can be performed several rounds to prevent the dishonest attempt of a prover. More specifically, for each round, a cheating prover can successfully pass the verification of the protocol with probability $\frac{1}{2}$. Nevertheless, if the prover and the verifier operate this protocol with polynomial iterations, then the dishonest prover can only pass the protocol with negligibly low probability, i.e., the verifier can be convinced

with high probability. Although the zero-knowledge watermark detection protocol can be used to show the presence of a watermark without needing to reveal it, the cheating behavior of a prover has not been avoided efficiently. As pointed out in [1], a cheating prover can intentionally choose a "faked" watermark as though it were the legal watermark to pass the verification of this protocol and deceive the verifier. The constraint is that the faked watermark should be chosen to be a detectable watermark still in the form of discrete logarithm. The existence of cheating prover in zero-knowledge watermark detection can be regarded as a variant of ambiguity attacks [6]. Although Craver [7] proposed a graph isomorphism and scrambling-based zero-knowledge watermark detection protocol, and exploited the hard problem of Hamiltonian cycle to prevent from the birthday attack, some difficulties (e.g., disclosing the characteristics of watermarks) still remain with this scheme. Another shortcoming of Craver's protocol is that a large number of iterative conversations between the prover and the verifier is required.

Another type of zero-knowledge watermark detection protocol, which exploits the existent zero-knowledge proof as its sub-function, is developed by Adelsbach and Sadeghi [1, 2]. For a watermarking scheme, the prover re-formulates the correlation between the original watermark and the extracted watermark into an appropriate form and employ the existing zero-knowledge protocol to prove for the verifier that the watermark exists without leaking any secret information. Once again, their protocol similar to Craver's protocol needs to be operated in a number of iterations. In [3], the authors further pointed out one particular requirement that in some applications the prover must prove in a zero-knowledge manner that her watermark must satisfy a certain distribution.

In addition to the above two identified problems (iterative communication and cheating prover), we also find that the zero-knowledge watermark detection protocols presented so far are not established based on a robust watermarking scheme. It is known that attacks were divided into four categories [20]: (1) removal attacks; (2) geometric attacks; (3) cryptographic attacks; and (4) protocol attacks. Therefore, motivated by the needs of sufficient robustness, this study focuses on proposing a non-interactive zero-knowledge watermark detection protocol based on the media hash-dependent watermarking schemes [14, 15], which were previously verified to be robust against extensive geometric attacks, and watermark-estimation attacks (including the collusion and copy attacks). It is quite practical to imagine that a watermark must be able to be detected from an attacked data by an owner in advance before the zero-knowledge watermark detection protocol is performed. Otherwise, the problem about "How can the owner prove to the verifier that he/she really owns the legal watermark, which unfortunately cannot be detected from a suspect data?" is unreasonable.

1.2 Contributions of this Work

In this paper, we study the following challenging problems: (i) zero-knowledge detection protocol is integrated with robust image watermarking schemes [14, 15] that were verified to be robust against extensive signal processing (including removal and geometric) attacks so that robustness is still retained; (ii) the watermarks revealed for zero-knowledge detection are still secured by a secret key-based shuffling function so that the verifier cannot have the knowledge about the hidden watermarks; (iii) Hamiltonian cycles are embedded into an image so as to prevent from dishonest prover; and (iv) we propose to package all the queries of the verifier and the corresponding answers of the prover in order to achieve non-interactive communication between the prover and the verifier. The above characteristics obviously distinguish our method between the existing methods.

2. MEDIA HASH-DEPENDENT IMAGE WATERMARKING SCHEME

Since the main theme of this study is to investigate a non-interactive zero-knowledge watermark detection protocol that is based on a robust watermarking scheme, it is necessary to start by briefly describing how our media hash-dependent watermarks are constructed. Then, the proposed protocol is described in the next section.

In this paper, our discussion of zero-knowledge watermark detection protocol will be build based on a mesh-based media hash-dependent image watermarking scheme [15] that has been verified to possess sufficiently robustness. Based on robust feature extraction and mesh generation processes, a cover image is first divided into a set of triangular meshes, $Mesh = \{M_i\}_{i=1,2,\cdots,M}$, where M denotes the number of meshes. For each mesh M_i, it is treated as an embedding unit and is embedded with a content-dependent watermark. In addition, with respect to the set of meshes $Mesh$ a set of media hashes $Hash = \{MH_i\}_{i=1,2\cdots M}$ is extracted and has been verified to be robust against extensive geometric attacks [17]. The proposed mesh-based media hash-dependent watermark $MMHW_i$ is composed of a watermark W generated using a secret key and a MH_i as

$$MMHW_i = S(MH_i \cdot W), \tag{1}$$

where S is a secret key-based shuffling function, which is used to control the combination of W and MH_i. In addition to resisting collusion attack and copy attack [14], we can also observe from Eq. (1) that error-resilient media hash instead of fragile cryptographic hash is adopted in this paper. Furthermore, our watermark W is not dependent on the cover image, instead W is made to (highly) correlated with the cover image by incorporating the media hash. Readers should refer to [15] for more details about the issues of how to construct a mesh, how a mesh-based media hash is generated, and how a watermark can be embedded into a mesh.

It is worth mentioning that since the proposed zero-knowledge watermark detection protocol is built based on this verified robust image watermarking scheme, resistance to signal processing attacks is still retained when ZKWD is mentioned. In the following, when we are talking about the zero-knowledge proof of a legal watermark, we refer to $MMHW_i$.

3. NON-INTERACTIVE ZERO-KNOWLEDGE WATERMARK DETECTION

The so-called non-interactive zero-knowledge (NIZK) proof stemmed from the idea of Santis *et al.* [18] and Blum *et al.* [4]. In the NIZK protocol, two parties do not need any communications, except sending one data set from the prover Alice to the verifier Bob [19]. This idea motivates us to

develop a non-interactive zero-knowledge watermark detection protocol. Moreover, a robust watermarking scheme [14, 15] is considered to be integrated with the proposed NIZK watermark detection protocol because only the watermarks can be successfully detected from suspect images, the NIZK detection protocol can be triggered. In what follows, the issues regarding media hash-dependent watermark embedding to retain robustness, construction of a Hamiltonian cycle to prevent from cheating provers, and non-interactive zero-knowledge detection protocol to avoid interactive conversations will be, respectively, described.

3.1 Media Hash-dependent Watermark Embedding

For image watermarking, Alice first publishes a prime number p and a positive integer $a \in [1, p-1]$, from which a watermark (W) is generated to be a discrete log form, i.e., $W \equiv a^x \pmod{p}$. Then, the media hash-dependent watermark is generated based on Eq. (1) and watermark embedding is performed using our previous method [15]. After these steps, let I^s denote the stego image. The watermark generation, the media hash-dependent watermark construction, and the embedding procedure are described as follows:

1. Alice chooses a positive integer x, sets $W \equiv a^x \pmod{p}$, and generates $MMHW_i = S(MH_i \cdot W)$ for the i-th mesh.
2. Let $MMHW = \{MMHW_i\}$ be denoted as the set of all media hash-dependent watermarks that will be embedded into a cover image I [15] and let I^s_{temp} temporarily denote the resultant stego image in this study.

In the proposed method, I^s_{temp} is further manipulated to contain so-called Hamiltonian cycles, as will be described in Sec. 3.2. After that, we let the Hamiltonian cycle embedded result be the final stego image that can be published and be denoted as I^s.

3.2 Use of Hamiltonian Cycle to Avoid Cheating Prover

In the ZKWD protocol, a cheating prover may construct a fake watermark still with a discrete log form and prove it to the verifier. In order to deal with this problem, for each image we randomly select a Hamiltonian cycle and embed [1] it into the image. Under this construction, the prover can present the Hamiltonian cycle of an image to prove that she indeed knows something secret about the stego image. Under this circumstance, a graph is said to have a Hamiltonian cycle if there exists a path passing through a graph that visits each node exactly once. On the other hand, if an attacker (e.g., cheating prover) tries to find the Hamiltonian cycle from an image, then it is necessary to evaluate where the Hamiltonian cycle is. According to [10], we know that this problem is NP-complete. In our study, this property will be exploited to prevent from cheating provers.

In what follows, the procedure of constructing a Hamiltonian cycle in an image is described.

1. Alice computes $I_{half} = MAP1(I^s_{temp})$, where the mapping function $MAP1$ transfers the entry $a_{i,j}$ of I^s_{temp} as the entry $b_{i,j}$ of I_{half} as

$$b_{i,j} = \begin{cases} 0, & i < j \text{ and } a_{i,j} \geq \texttt{THRESHOLD} \\ 1, & i < j \text{ and } a_{i,j} \leq \texttt{THRESHOLD} - 1 \\ 0, & i = j \\ a_{i,j}, & i > j. \end{cases}$$

   This step is used to transfer a gray-scale image I^s_{temp} into another image I_{half}, whose upper-triangle matrix contains binary entries.

2. Alice sets $I_{graph} = MAP2(I_{half})$, where the entry $c_{i,j}$ of I_{graph} and the entry $b_{i,j}$ of I_{half} are related as

$$c_{i,j} = \begin{cases} b_{i,j}, & i \leq j \\ b_{j,i}, & i > j. \end{cases}$$

   After this step, the image I_{half} is transferred to a graph I_{graph}, where the value of entry denotes the existence of an edge linking between two nodes; i.e., $c_{i,j} = 1$ denotes that there is an edges between the nodes i and j.

3. Alice makes $I^{HC}_{graph} = adj(I_{graph})$, where the function $adj()$ is used to construct a Hamiltonian cycle by modifying I_{graph} as I^{HC}_{graph}. Let the entries of I^{HC}_{graph} be denoted as $d_{i,j}$'s. In this study, we define $adj()$ as follows. For an image of size $N \times N$, we randomly select a Hamiltonian matrix $HM|_{N \times N}$, which is an undirected graph with N vertices and N edges, and the total edges constitute a Hamiltonian cycle. Then, $adj(I_{graph})$ is rewritten as $adj(I_{graph}) = I_{graph} \vee HM$, where $\vee$ denotes the OR operation. We will further discuss possible attacks on the construction of Hamiltonian cycles in Sec. 3.2.3.

4. Alice sets $I^s = MAP3(I^s_{temp}, I_{graph}, I^{HC}_{graph})$, where the mapping function is used to relate the entry $e_{i,j}$ of I^s and the entries of I^s_{temp}, I_{graph}, and I^{HC}_{graph} as

$$e_{i,j} = \begin{cases} a_{i,j}, & i < j \text{ and } c_{i,j} = d_{i,j} \\ \texttt{THRESHOLD} - 1, & i < j \text{ and } c_{i,j} < d_{i,j} \\ \texttt{THRESHOLD}, & i < j \text{ and } c_{i,j} > d_{i,j} \\ a_{i,j}, & i \geq j. \end{cases}$$

   In this step, the design of $e_{i,j}$'s in the second and the third rows directly affect the robustness of Hamiltonian cycles and the fidelity of a stego image. These issues will be further discussed in the remainder of this paper.

As described in the above procedure, the functions, $MAP1$ (), $MAP2()$, $MAP3()$, and $adj()$, are used to transform the temporary stego image I^s_{temp} into a publishable stego image I^s by slightly modifying I^s_{temp} to set up an invisible Hamiltonian cycle. It should be noted that by comparing the differences between I^{HC}_{graph} and I_{graph}, we can discover which pixels of I^s_{temp} have to be modified by referring both I_{graph} and I^{HC}_{graph} to yield the stego image I^s. It is not hard to find that all these steps can be finished in polynomial time that is proportional to the size of an image. In order to better illustrate the procedure of constructing a Hamiltonian cycle in an image, a toy example is further described in Sec. 3.2.1.

Without loss of generality, THRESHOLD is set to 128 by assuming pixel values are a uniform distribution. Nevertheless, the distribution of pixel values is not always uniform.

[1] It should be noted that unlike [7] we propose an alternative way of using Hamiltonian cycles to deal with the problem of cheating provers.

As a result, we can use the average pixel value as `THRESHOLD`. Under this circumstance, when the average pixel value is small or large enough, it is possible for a cheating prover to easily execute exhaustive search so as to find the Hamiltonian cycle in an image.

However, embedding of a Hamiltonian cycle is equivalent to embedding of a watermarking in that the conflicting requirements of invisibility and robustness must be satisfied. Although the above simple implementation cannot fulfill this goal, embedding technologies exist to achieve it. This is because we also find that the role of a Hamiltonian cycle in resisting the cheating behaviors of provers is equivalent to that of a pilot signal in recovering the asynchronization induced by geometric attacks. Since the pilot signals is easily defeated by means of the collusion and copy attacks [14], the desired functionality is lost as well. Similarly, if the robustness of Hamiltonian cycles cannot achieve the same level as watermarks, the functionality of resisting cheating provers will also be lost. Fortunately, we have presented a robust image watermarking scheme [15] that can prevent from estimation attacks. As a result, we will plan to enhance the fidelity and robustness of embedded Hamiltonian cycles by exploiting the scheme [15] instead of the simple implementation illustrated here.

On the other hand, if the mesh-based media hash-dependent watermarking scheme [15] is considered, the above procedure can be similarly applied to each mesh so that multiple Hamiltonian cycles are embedded into an image. Under these circumstances, to find a Hamiltonian cycle from an image will become more difficult, as described in Sec. 3.2.3.

3.2.1 A Toy Example of Constructing Hamiltonian Cycle

Let the temporary stego image I^s_{temp} be simply expressed as

$$I^s_{temp} = \begin{pmatrix} 200 & 193 & 129 & 244 & 20 \\ 234 & 223 & 182 & 22 & 209 \\ 245 & 93 & 239 & 173 & 27 \\ 150 & 23 & 188 & 237 & 200 \\ 19 & 175 & 10 & 263 & 190 \end{pmatrix},$$

which is a 5×5 matrix and its entries denote the pixel values. According to the first two steps of Hamiltonian cycle construction procedure (Sec. 3.2), I_{graph} can be obtained by applying the functions, $MAP1()$ and $MAP2()$, on I^s_{temp} as

$$I_{graph} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Then, in the third step we select a Hamiltonian matrix HM as

$$HM = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

By means of applying the OR operation between I_{graph} and HM, we derive

$$I^{HC}_{graph} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \vee \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

In the fourth step, the stego image I^s that can published is derived as

$$I^s = \begin{pmatrix} 200 & 127 & 127 & 244 & 20 \\ 127 & 223 & 182 & 22 & 209 \\ 127 & 93 & 239 & 173 & 27 \\ 150 & 23 & 188 & 237 & 127 \\ 19 & 175 & 10 & 127 & 190 \end{pmatrix}.$$

It should be noted that we have not concerned the distribution of pixel values and the threshold $THRESHOLD = 128$ is simply used.

3.2.2 Embedding of a Hamiltonian Cycle vs. Fidelity

This issue is discussed from two cases. In the first case, the corresponding graph I_{graph} of a stego image of size $N \times N$ obtained from the first two steps of Sec. 3.2 contains no edges. This represents the worst case that more pixels are needed to be modified to construct a Hamiltonian cycle. Even so, it is adequate to modify exactly N pixels to guarantee the generation of a Hamiltonian cycle.

In the second case, we consider that I_{graph} initially contains edges. To simplify analysis, it is assumed that the distribution of the image is uniform and we would like to know how many pixels of an image need to be modified to create a Hamiltonian cycle. For an image of size $N \times N$, there are at most $\binom{N}{2}$ edges. Based on the assumption that the pixel values form a uniform distribution, an image, without loss of generality, is considered to form a graph having N nodes and $\lfloor \frac{\binom{N}{2}}{2} \rfloor$ edges. On the other hand, let a complete subgraph be composed of $\lceil \frac{N}{2} \rceil$ nodes and $\binom{\lceil \frac{N}{2} \rceil}{2}$ edges. For any N, we can derive

$$\lfloor \frac{\binom{N}{2}}{2} \rfloor > \binom{\lceil \frac{N}{2} \rceil}{2},$$

which means that there exists at least one edge linking this $\lceil \frac{N}{2} \rceil$-node complete subgraph to one of the other $\lfloor \frac{N}{2} \rfloor$ nodes. Since a Hamiltonian cycle is a sequence of N edges passing through all vertices in a graph, therefore, it is sufficient for us to modify *at most* $N - ((\lceil \frac{N}{2} \rceil) - 1 + 1) = \lfloor \frac{N}{2} \rfloor$ pixels to derive a graph that contains a Hamiltonian cycle. In order to enhance the practicality of Hamiltonian cycle embedding, both watermarks and Hamiltonian cycles should be embedded in the same way because Hamiltonian cycles act like watermark signals and must satisfy invisibility and robustness against attacks.

3.2.3 Attacks on Construction of Hamiltonian Cycles

In this study, the embedding of Hamiltonian cycles is introduced to prevent cheating prover. A concern here is that

the dishonest prover may try to find the Hamiltonian cycles by exploiting the existing techniques (e.g., Bollobás *et al.*'s algorithm [5]). In the following, we will discuss the possibility of detecting the embedded Hamiltonian cycles.

Let $G_{v,\kappa}$ be a graph with v vertices and κ edges. In 1985, Bollobás *et al.* designed a polynomial-time randomized algorithm for finding Hamiltonian cycles in a graph $G_{v,\kappa}$ with the constraint that $\kappa = \frac{v \log v}{2} + \frac{v \log \log v}{2} + \delta v$ and δ is a real number. The probability of finding Hamiltonian cycles in a graph is derived [5] as

$$\lim_{v \to \infty} (\text{find a hamiltonian cycle}) = \begin{cases} 0, & \delta \to -\infty \\ e^{-e^{-2c}}, & \delta \to C \\ 1, & \delta \to \infty, \end{cases}$$

where C is a constant. At the first glance, it seems that Bollobás *et al.*'s algorithm can be used to detect the existence of Hamiltonian cycles. Nevertheless, their algorithm can only hold theoretically when the number of vertices approaches an infinite number, which is impractical. In addition, when the number of edges in a graph is smaller than $\frac{v \log v}{2} + \frac{v \log \log v}{2} + 0.1v$, i.e., when $\delta = 0.1$, the probability defined in Eq. (2) is approximate 0.44. Thus, it is important to select a proper value of $THRESHOLD$ for use in Sec. 3.2. Furthermore, if several Hamiltonian cycles (the number depends on the number of meshes in an image in our study) are constructed for an image, the probability for a dishonest prover to find all the embedded Hamiltonian cycles would be sufficiently low (e.g., the probability would be $0.44^{25} \approx 1.22e^{-9}$ if 25 Hamiltonian cycles are constructed). Thus, we need a solution to defeat Bollobás *et al.*'s algorithm in finding Hamiltonian cycles. In view of these, it becomes obviously that the design of constructing Hamiltonian cycles in an image can be exactly consistent with the employed image watermarking scheme, as described in Sec. 2. This is because our watermarking scheme is proposed to be mesh-based and each mesh area can be embedded with a Hamiltonian cycle with the aim that more than one Hamiltonian cycle is embedded.

3.3 Non-Interactive Zero-Knowledge Watermark Detection

According to the mesh-based media hash-dependent image watermarking scheme [15], a stego image is assumed to be decomposed into $\acute{\alpha}$ meshes and α of them can be detected by means of our watermark detection process (which means that the detected correlation is larger than a pre-determined threshold), where $\acute{\alpha} \geq \alpha$. Now, the proposed non-interactive watermark detection protocol is described as follows.

1. At the beginning, Alice sets $F = \{F^1, \cdots, MMHW, \cdots, F^n\}$, where $MMHW = F^t$ $(1 \leq t \leq n)$ and the cardinality $|F^j|$ of the set F^j is α. Let F^j be represented as $\{f_k^j | 1 \leq k \leq \alpha\}$. Among the elements of F^j $(j \neq t)$, they do not contain the originally embedded watermarks; on the contrary, all the watermarks except for those in $MMHW$ that can be detected can be easily generated by means of denoising-based watermark extraction [14] or ambiguity attack [6]. For example, one can use some high-pass filters (e.g., Wiener filtering) to filter the stego image so that a set of extracted watermarks can be yielded.

2. Alice randomly chooses m positive integer matrix of size $\alpha \times n$, where the entry of i-th matrix is $a^{y_{j,k}^i}$ and $1 \leq i \leq m$, $1 \leq j \leq n$, $1 \leq k \leq \alpha$. Here, m denotes the number of interactive conversations used in a traditional interactive zero-knowledge proof protocol. Then, each element of F is multiplied by the integer matrix to yield the corresponding set $W^i = \{W_1^i, \cdots, W_t^i, \cdots, W_n^i\}$ for $1 \leq i \leq m$, where W_t^i is derived from $MMHW$. Let W be denoted as $\{W^i | 1 \leq i \leq m\}$.

3. Alice devises a set of graph isomorphism corresponding to I_{graph}^{HC}, $G_{iso} = \{G_1, \cdots, G_m\}$, and encrypts them into $G'_{iso} = \{G'_1, \cdots, G'_m\}$. For graph isomorphism, we mean that the nodes are randomly permuted and their edges are also permuted accordingly. In addition, the graph encryption considered here is only operated on the edges of a given graph. Graph isomorphism can be done in polynomial-time since the number of swap operations is proportional to the number of nodes. The aim of this step [19] is for the prover to prove to the verifier that she knows the Hamiltonian cycles of I_{graph}^{HC} without revealing them, as will be described in step 5.

4. In order to achieve non-interactive conversation, Alice has to prepare a sequence Q of random queries. To do so, the random queries can be yielded by means of one way hashing, i.e., $Q = q_1 q_2 \cdots q_m = \mathbf{OWH}(I^s)$. The function **OWH** is public so that the verifier Bob can check whether the query sequence Q is randomly generated from the publicly known stego data I^s.

5. According to the value of q_i, the prover Alice prepares the set $A = \{A_i | 1 \leq i \leq m\}$ of answers in advance:

   - if $q_i = 0$, Alice must show the relation between F^i and W^i by revealing the corresponding integers $y_{j,k}^i$'s for all $1 \leq j \leq n$ and $1 \leq k \leq \alpha$. In addition, Alice also records the vertices permutation between G_i and I_{graph}^{HC} without showing their Hamiltonian cycles. All the above information is represented as a set A_i, which is revealed to Bob.
   - if $q_i = 1$, Alice must give Bob $x + y_{j,k}^i$'s for $j = t$ and $1 \leq k \leq \alpha$. In addition, Alice also needs to let Bob know the Hamming distance between $a^{x+y_{t,k}^i}$ and the corresponding elements in $F^t (= MMHW)$, and the Hamiltonian cycle of G_i by decrypting only the edges in the cycle from G'_i without showing the topological isomorphism between G_i and I_{graph}^{HC}. Again, all the above information is represented as a set A_i, which is revealed to Bob.

6. Finally, Alice sends the certificate, $\{F, W, Q, G_{iso}, G'_{iso}, A\}$ altogether to Bob. It should be noted that the exposed set of watermarks, which are designed based on Eq. (1), is still protected by means of the shuffling function S.

In our protocol, the full understanding of a Hamiltonian cycle in an image is exploited to distinguish true prover between false prover. Also, one may concern "whether some secrets may be leaked?" The answer is "NO" because in each round i the prover gives the verifier either the vertex permutation between the I_{graph}^{HC} and G_i or the Hamiltonian cycle of G_i by decrypting G'_i. This implies the prover never gives both of them to the verifier. As a consequence, the

verifier cannot get the whole knowledge of the Hamiltonian cycle in I^s.

The principle behind our non-interactive zero-knowledge proof is to package all the answers (as indicated in step 6) that the prover requires to prepare and all the queries that the verifier will raise into a set. In order for the proposed protocol to be fair and practical, the sequence of queries is generated by means of a cryptographic one-way hash function with the stego image I^s as the input, as indicated in step 4. In the other words, the i-th bit of $Q = q_1 q_2 \cdots q_m = \mathbf{OWH}(I^s)$ is equal to the query of the verifier in i-th round of zero-knowledge proof. Although the overhead of information that should be transmitted from the prover to the verifier is not reduced, the number of interactions has been greatly reduced to only one. The additional merit is that the verifier can check the copyright of media data offline.

Despite the more or less trivial interactive protocol presented by us, it would be fair to say that the existing protocols can also be made non-interactive using other designs.

3.4 Remarks

It can be found that our non-interactive zero-knowledge watermark detection protocol follows the principle proposed by Craver [7, 8]. However, our protocol presents some characteristics that distinguish ours, Craver's, other methods [1, 2, 3]. First, the aim of our detection protocol is designed to extend our robust watermarking scheme [15] that has been verified to resist removal attacks, extensive geometric attacks, and watermark-estimation attacks [14] to further satisfy zero-knowledge watermark detection. Second, our protocol is designed to operate in a non-interactive manner such that the major weakness of zero-knowledge detection is alleviated. Third, the set of originally embedded watermarks, F^t, that is sent out publicly are still secured except that the shuffling function [16] described in the first step 1 of Sec. 3.1 can be easily broken. These characteristics are quite different from the previous work in the literature.

4. RELATIONSHIP BETWEEN HONEST/CHEATING PROVER AND HONEST/CHEATING VERIFIER

In the zero-knowledge detection protocol, conversations happen between the prover and the verifier so that the prover can prove to the verifier that she really knows the hidden watermark. In practice, both the prover and the verifier may be honest or dishonest, as shown in Fig. 1, where four scenarios may occur depending on the behaviors of the prover and verifier. Figs. 1(a) and (b) show the abilities of current protocols and our protocol in dealing with these four scenarios, each of which will be explained in the following.

First, let's consider that both the prover and verifier are honest in that the verifier has a stego image legally obtained from the owner, and the prover legally represents the owner to execute zero-knowledge watermark detection. According to this scenario, the prover can exploit either the existing protocols or our protocol to successfully prove the existence of a watermark to the verifier.

In the second scenario, we consider that both the prover and the verifier are dishonest. This is an interesting problem because the verifier can attack the received stego image I^s to destroy/remove the hidden watermark, and the cheating prover has no knowledge about the to be detected image

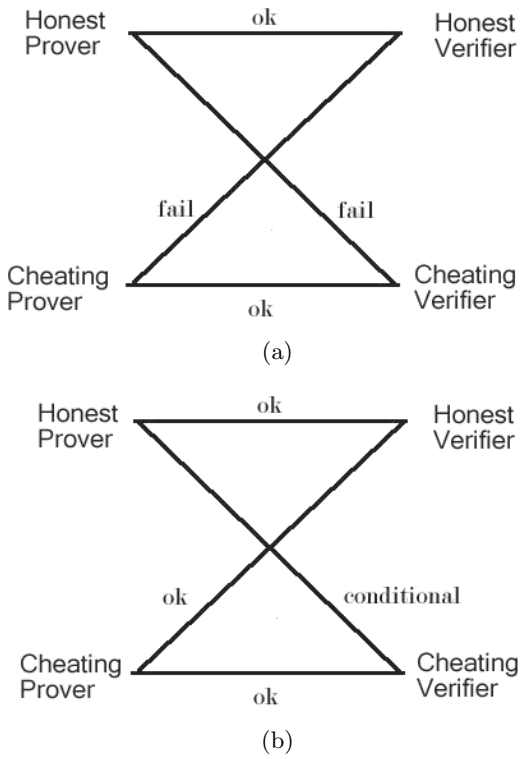

Figure 1: Relationship between the honest/dishonest prover and the honest/dishonest verifier: (a) the ability of current protocols; (b) the ability of our protocol.

and tries to defraud the verifier. Obviously, a protocol is not required to be responsible for both dishonest parties, and this scenario can be ignored.

Third, when the cheating prover, who can fake the watermark, meets the honest verifier, the current zero-knowledge protocols suffer from this challenging problem. However, the introduced proof of the Hamiltonian cycle in an image can prevent from the cheating prover since we show that the cheating prover always cannot find Hamiltonian cycles from an image in reasonable time. Therefore, the third scenario is approximately solved by our protocol.

As for the fourth scenario, the honest prover meets the dishonest verifier, who will attack the received stego image. We consider that the solution to this scenario needs a *robust* zero-knowledge watermark detection protocol. If a zero-knowledge detection protocol is not established based on a robust watermark scheme, we consider that the watermarks cannot be successfully detected before the ZKWD protocol can be applied. Although the proposed protocol has been built on a robust image watermark scheme that was verified to be robust against benchmark attacks, the introduced Hamiltonian cycles, in its current status, may not be able to resist attacks (e.g., achieve the same level of robustness for watermarks to resist attacks). Nevertheless, if the partially recovered Hamiltonian cycle due to attacks is used, this problem may be properly solved in certain conditions. In other words, the robustness of a Hamiltonian cycle against attacks plays a key role, and this issue should be further studied.

5. CONCLUSIONS

Zero-knowledge watermark detection is practically important for watermark verification without (obviously) revealing any secret information. In this paper, we investigate this problem with particular emphasis on the issues of non-interactive conversation, prevention from cheating prover, and retaining robustness against attacks. Due to the non-interactive characteristic of our protocol, the prover Alice outputs publicly a certificate for zero-knowledge watermark detection so that we may regard this kind of non-interactive zero-knowledge detection protocol as a kind of asymmetric watermarking.

While zero-knowledge watermark detection protocol is an important step towards secure watermark detection, it can only function reasonably if an embedded watermark can be detected/extracted from attacked media data. As a result, robustness against attacks is believed to be the prerequisite that must be satisfied before zero-knowledge watermark detection protocol can be applied. That's why we attempt to extend the capability of our previous robust watermarking scheme to cover zero-knowledge watermark detection.

A problem we would like to particularly point out is resistance to ambiguity attacks (e.g., a kind of protocol attacks [20]), which according to Craver *et al.*'s protocol [6] requires the owner to show how the so-called "original" watermark is generated from the so-called "original" cover data. Thus, the secret information is publicly revealed. Therefore, it is noted that the requirements of resistance to protocol attacks and zero-knowledge watermark detection protocol are contradictory. We think this is an interesting direction for further researching.

Although the proposed ZKWD protocol has been built on a robust image watermarking scheme, there are other robustness issues required for further researching in order to complete a robust ZKWD protocol. These include (i) robustness of the embedded discrete-form watermarks; and (ii) robustness of the embedded Hamiltonian cycles. One possible solution to the latter case is to employ partial matching so that the existence of a Hamiltonian cycle can still be measured. In practice, the current paper assumes that the stego image is the target for zero-knowledge watermark detection in order to avoid the above issues.

Acknowledgment: This research was supported, in part, by the National Science Council under NSC grant 93-2422-H-001-004.

6. REFERENCES

[1] A. Adelsbach and A. Sadeghi, "Zero-Knowledge Watermark Detection and Proof of Ownership," *Proc. Int. Workshop on Information Hiding*, LNCS 2137, pp. 273-288, 2001.

[2] A. Adelsbach, S. Katzenbeisser, and A. Sadeghi, "Watermark Detection with Zero-Knowledge Disclosure," *ACM Multimedia Systems Journal*, Vol. 9, No. 3, pp. 266-278, 2003.

[3] A. Adelsbach, M. Rohe, and A. Sadeghi, "Overcoming the Obstacles of Zero-Knowledge Watermark Detection," *Proc. ACM Multimedia and Security Workshop*, Magdeburg, Germany, pp. 46-55, 2004.

[4] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," *Proc. of the 20th ACM Symposium on Theory of Computing*, pp. 103-112, 1988.

[5] B. Bollobás, T I, Fenner, and A M. Fireze, "An Algorithm for Finding Hamilton Cycles in a Random Graph," *Proc. of the 17th ACM Symposium on Theory of Computing*, pp. 430-439, 1985.

[6] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeng, "Resolving Rightful Ownership with Invisible Watermarking Techniques: Limitations, Attacks, and Implications," *IEEE Journal on Selected Areas in Comm.*, Vol. 16, No. 4, pp. 573-586, 1998.

[7] S. Craver, "Zero Knowledge Watermark Detection," *Proc. Int. Workshop on Information Hiding*, LNCS 1768, pp. 107-122, 1999.

[8] S. Craver, B. Liu, and W. Wolf, "An Implementation of, and Attacks on, Zero-Knowledge Watermarking," *Proc. Int. Workshop on Information Hiding*, LNCS 3200, pp. 1-12, 2004.

[9] J. J. Eggers, J. K. Su, and B. Girod, "Asymmetric Watermarking Schemes," *Sicherheit in Netzen und Medienstromen*, Springer Reihe, Infomatick Aktuell, 2000.

[10] M.R. Garey, D.S. Johnson, and L. Stockmeyer, "Some Simplified NP-Complete Problems," *Proc. of the sixth ACM Symposium on Theory of Computing*, pp.47-63, 1974.

[11] G. Hachez and J-J. Quisquater, "Which Directions for Asymmetric Watermarking?" *Proc. of EUSIPCO*, France, 2002.

[12] F. Hartung and B. Girod, "Fast Public-Key Watermarking of Compressed Video," *Proc. IEEE Int. Conf. on Image Processing*, 1997.

[13] A. Kerckhoffs, "La cryptographie mililaire," *J. Sci. Militaires*, Vol. 9, pp. 5-38, 1883.

[14] C. S. Lu and C.Y. Hsu, "Content-Dependent Anti-Disclosure Image Watermark," *Proc. 2nd Int. Workshop on Digital Watermarking*, LNCS 2939, pp. 61-76, Seoul, Korea, 2003.

[15] C. S. Lu, S. W. Sun, and P. C. Chang, "Robust Mesh-based Content-dependent Image Watermarking with Resistance to Both Geometric Attack and Watermark-Estimation Attack," *Proc. SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII (EI120)*, pp. 147-163, San Jose, California, USA, 2005. (The full version is available as a technical report (TR-IIS-05-002) at http://www.iis.sinica.edu.tw/LIB/TechReport/tr2005/threebone05.html)

[16] C. S. Lu and C. M. Yu, "On the Security of Mesh-based Media Hash-dependent Watermarking Against Protocol Attacks," *Proc. IEEE Int. Conf. on Multimedia and Expo*, The Netherlands, 2005.

[17] C. S. Lu and C. Y. Hsu, "Geometric Distortion-Resilient Image Hashing Scheme and Its Applications on Copy Detection and Authentication," accepted and to appear in *ACM Multimedia Systems Journal*, special issue on Multimedia and Security (preliminary version was published in *Proc. ACM Multimedia and Security Workshop*, pp. 81-92, Magdeburg, Germany, 2004).

[18] A. D. Santis, S. Micali, and G. Persiano, "Non-Interactive Zero-Knowledge Proof Systems,"

Advances in Cryptology - CRYPTO '87 Proceedings, pp. 52-72, 1987.

[19] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source code in C," 2nd ed., John Wiley and Sons, 1996.

[20] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack Modeling: Towards a Second Generation Watermarking Benchmark," *Signal Processing*, Vol. 81, pp. 1177-1214, 2001.